IN THE CLAIMS:

Please cancel claims 1 to 51.

Claims 52 and 53 are unchanged. They are reproduced here for the Examiner's convenience:

52. A method for recording a plurality of data about a plurality of blocks of data stored in storage means, comprising the steps of:

maintaining a means for recording multiple usage bits per block of said storage means; and

storing, in said means for recording multiple usage bits per block, multiple bits for each of said plurality of said blocks of said storage means, at least one of said multiple bits being indicative of block reusability.

53. A method for recording a plurality of data about a plurality of blocks of stored data, comprising the steps of:

recording multiple usage bits per block of said stored data; and

storing multiple bits for each of said plurality of said blocks of stored data, at least one of said multiple bits being indicative of block reusability.

Please add claims 54 to 147 as follows:

54. A file system, having a plurality of storage blocks, and including a plurality of bits associated with each one of said plurality of storage blocks, at least one of said plurality of bits identifying whether said one storage block was part of said file system at a time earlier than a current consistent version of said file system.

55. A file system as in claim 54, including a second one of said plurality of bits identifying whether said one storage block was part of said file system at a second time earlier than a current consistent version of said file system

A1 Cm.1

56. A file system as in claim 55, including an element disposed for selecting storage blocks in response to said one bit and said second one bit associated with said selected storage blocks.

57. A file system as in claim 56, including an element disposed for copying said selected storage blocks to a destination.

58. A file system as in claim 57, wherein said destination includes: a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

59. A file system as in claim 54, including an element disposed for selecting storage blocks in response to said one bit associated with said selected storage blocks.

60. A file system as in claim 59, including an element disposed for copying said selected storage blocks to a destination.

A1 Cm.1

61. A file system as in claim 60, wherein said destination includes: a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

62. A file system having a plurality of storage blocks, said file system including a snapshot including a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system; said snapshot being disposed as an object in said file system, wherein said file system is responsive to at least one file system request with regard to said snapshot.

63. A file system as in claim 62, including

a mark-on-allocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been added to said snapshot; and

a storage image defined based on said snapshot and said mark-on-allocate image, said storage image indicating a set of member storage blocks selected from said plurality.

64. A file system as in claim 63, wherein said storage image is defined with regard to a logical sum operation on said snapshot and said mark-on-allocate image.

65. A file system as in claim 62, including

a mark-on-deallocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been removed from said snapshot; and

a storage image defined based on said snapshot and said mark-on-deallocate image, said storage image indicating a set of member storage blocks selected from said plurality.

A1 cm.1

66. A file system as in claim 62, including

a shadow snapshot of a set of member storage blocks selected from said plurality, said member storage blocks having formed a consistent file system other than an active file system, with a set of selected member storage blocks removed from said consistent file system; and

a storage image defined based on said snapshot and said shadow snapshot, said storage image indicating a set of member storage blocks selected from said plurality.

67. A file system as in claim 62, including an indicator of which ones of said member storage blocks have been copied.

68. A file system as in claim 62, including a plurality of said snapshots; wherein said plurality of said snapshots are associated with an array of bits, said array having one set of bits for each storage block in said plurality of storage blocks, said set of bits having at least one bit for each said snapshot.

69. A file system as in claim 62, wherein said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

A1 Cm.1

70. A file system as in claim 62, wherein said snapshot includes a data structure disposed in a format allowing for a set management operation to be performed efficiently.

71. A file system as in claim 62, wherein said snapshot includes an array of bits, said array having one bit for each storage block in said plurality.

72. A file system as in claim 62, including

a plurality of said snapshots; and

a storage image determined based on said plurality of snapshots;

said storage image defining a second set of member storage blocks selected from said plurality.

73. A file system as in claim 72, including an indicator of which ones of said storage blocks in said storage image have been copied.

74. A file system as in claim 72, wherein said storage image is a result of a logical sum or difference performed on said set of member storage blocks for said snapshot and a set of member storage blocks for a second said snapshot.

A1 cmx

75. A file system as in claim 72, wherein said storage image is a result of a logical sum or difference performed on said set of member storage blocks for said snapshot and a set of member storage blocks for a second said storage image.

76. A file system as in claim 72, wherein said storage image is a result of a set management operation on said set of member storage blocks for said snapshot.

77. A file system as in claim 62, wherein said snapshot includes a data structure disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

78. A file system as in claim 77, wherein said set management operation is a logical sum or difference.

79. A file system as in claim 62, wherein said snapshot includes a data structure identifying which storage blocks in said plurality are member storage blocks of said snapshot.

80. A file system as in claim 79, wherein said data structure uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

81. A file system as in claim 79, wherein said data structure uses no more than four bytes per storage block in said plurality.

A1 cm.X

82. A method of operating a file server, said method including steps for

forming a first snapshot of a first consistent state of said file system at a selected time, said first snapshot including an indication of a set of storage blocks in said first consistent state;

forming a second snapshot of a second consistent state of said file system, said second snapshot including an indication of a set of storage blocks in said second consistent state; and

performing an operation on said first and second snapshots to form a storage image including an indication of at least some storage blocks in said file system.

83. A method as in claim 82, wherein said operation includes a logical sum or difference.

84. A method as in claim 82, wherein

said operation includes a logical sum or difference; and

a purpose of said operation includes making a copy including or excluding a selected range of snapshots.

85. A method as in claim 82, wherein

said operation includes a logical sum or difference; and

a purpose of said operation includes copying said storage image to a destination.

86. A method as in claim 85, wherein said destination includes a tape, a disk, a data structure in a second file system, a set of network messages, or a destination distributed over a plurality of file systems.

87. A method to be performed in a file system, said file system having a plurality of storage blocks, said method including steps for

defining a storage image of a set of member storage blocks selected from said plurality, said storage image being formed based on a set of member storage blocks forming a consistent file system other than an active file system; and

forming an image stream of a sequence of member storage blocks selected from said storage image.

88. A method as in claim 87, including steps for associating a block location with each one of said sequence.

89. A method as in claim 87, further including steps for reconstructing a file system based on said image stream.

90. A method as in claim 87, wherein said steps for forming are performed in response to a selected operation to be performed on said member storage blocks, said selected operation being other than an operation on an active file system

91. A method as in claim 87, wherein said steps for forming include steps for optimizing said sequence of member storage blocks for a file system operation.

92. A method as in claim 87, wherein said steps for forming include steps for optimizing said sequence of member storage blocks for a file system operation in a RAID file system.

93. A method as in claim 87, wherein said steps for forming include steps for optimizing said sequence of member storage blocks based on a physical location in a storage medium for each said member storage block, said storage medium having a plurality of storage elements capable of being read in parallel; and

ordering said sequence of member storage blocks so that said member storage blocks can be optimally read in parallel from said plurality of storage elements.

94. A method as in claim 87, wherein said storage image represents a complete file system.

Al Cont

95. A method as in claim 87, wherein said storage image represents a set of changes to a file system.

96. A method as in claim 87, including repeating said defining step at periodic intervals.

97. A method as in claim 87, including repeating said defining step in response to an operator command.

98. A method as in claim 87, including repeating said selecting step in response to a remote device.

99. An incremental mirror copy of a file system, said incremental copy including a base set of storage blocks stored in a first storage medium, and an incremental set of storage blocks stored in a second storage medium.

100. An incremental mirror copy as in claim 99, wherein

said first storage medium is slower than said second storage medium.

101. An incremental mirror copy as in claim 99, wherein said incremental set of storage blocks is responsive to a plurality of updates of said file system.

102. An incremental mirror copy as in claim 99, wherein said incremental set of storage blocks is responsive to a continuous sequence of updates of said file system, wherein said incremental mirror copy includes an up to date set of storage blocks in said file system.

A1 Cn.X

103. An incremental mirror copy as in claim 99, wherein said incremental set of storage blocks is responsive to an indication of a set of storage blocks deallocated from said file system.

104. Apparatus including

a file system including a plurality of snapshots thereof, each representing an associated consistent state at an associated selected time; and

each said snapshot including an indication of a set of storage blocks in said associated consistent state, said indication being recorded in at least one storage block in said associated consistent state.

105. Apparatus as in claim 104, including a storage image defining at least some storage blocks in said file system, said storage image based on an operation on at least two of said snapshots.

106. An incremental mirror of a file system having a plurality of storage blocks, said incremental mirror including

a first set of first member storage blocks selected from said plurality, said first member storage blocks forming a copy of a first consistent version of said file system; and

a second set of second member storage blocks selected from said plurality, said second member storage blocks being based on said first consistent version and on a second consistent version of said file system, said second set including a set of changes between said first and second consistent version;

said first set being stored in a first storage medium, and said second set being stored in a second storage medium of different type;

whereby a complete copy of said file system can be constructed from said first set and said second set.

107. An incremental mirror as in claim 106, wherein said first storage medium has greater storage capacity and is slower than said second storage medium.

108. An incremental mirror copy as in claim 106, wherein said second set of member storage blocks is responsive to a plurality of updates of said file system.

109. An incremental mirror copy as in claim 106, wherein said second set of member storage blocks is responsive to a continuous sequence of updates of said file system, wherein said second set of member storage blocks includes a up to date set of storage blocks in said file system.

A1 Cont

110. An incremental mirror copy as in claim 106, wherein said second set of member storage blocks is responsive to an indication of a set of storage blocks deallocated from said file system.

111. In a file system having a plurality of storage blocks, a data structure including

a first snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said first snapshot being represented as an object in said file system and having a set of storage blocks for recording said first snapshot;

whereby copying said member storage blocks in said first snapshot has the property of preserving at least one snapshot recorded in said file system at a time of said first snapshot.

112. A data structure as in claim 111, including

a second snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said second snapshot being represented as an object in said file system and having a set of storage blocks for recording said second snapshot;

whereby copying said member storage blocks in said second snapshot has the property of preserving at least one snapshot recorded in said file system at a time of said second snapshot.

Al Cm.X

113. A data structure as in claim 111, including

an image stream including a set of storage blocks including both said first snapshot and said second snapshot;

whereby copying said member storage blocks in said image stream has the property of preserving both said first snapshot and said second snapshot.

114. In a file system having a plurality of storage blocks, a data structure including

a snapshot of a set of member storage blocks selected from said plurality, said member storage blocks forming a consistent file system other than an active file system;

said snapshot being represented as an object in said file system and having a set of storage blocks for recording said snapshot;

whereby a backup and restore operation on said file system has the property of preserving said snapshot within said file system.

115. In a file system having a plurality of storage blocks, a data structure including

a storage image of a set of member storage blocks selected from said plurality;

said storage image being formed based on a set of member storage blocks forming a consistent file system other than an active file system.

Al cm.x

116. A data structure as in claim 115, including

a first storage image indicating a set of member storage blocks forming a consistent file system; and

a sequence of incremental storage images, each having a predecessor, at least one of said predecessors being said first storage image.

117. A data structure as in claim 115, including an indicator of which ones of said storage blocks in said storage image have been copied.

118. A data structure as in claim 115, wherein said storage image indicates a logical difference of two sets of member storage blocks, at least one of said sets forming a consistent file system.

119. A data structure as in claim 115, wherein said storage image indicates a logical sum of two sets of member storage blocks each collectively forming a consistent file system.

120. A data structure as in claim 115, wherein said storage image indicates a set of member storage blocks forming a consistent file system.

A1 Cm.1

121. In a file system having a plurality of storage blocks, a data structure stored in said file system, including a shadow snapshot of a set of member storage blocks selected from said plurality, said member storage blocks having formed a consistent file system other than an active file system, with a set of selected member storage blocks removed from said consistent file system.

122. A data structure as in claim 121, wherein said shadow snapshot is disposed in a format allowing for a set management operation to be performed efficiently.

123. A data structure as in claim 121, wherein said shadow snapshot uses, in addition to said member storage blocks, no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

124. A data structure as in claim 121, wherein said shadow snapshot uses, in addition to said member storage blocks, no more than one byte per storage block in said plurality.

125. A data structure as in claim 121, wherein said shadow snapshot is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

A1 Cm X

126. A data structure as in claim 121, wherein said removed member storage blocks are responsive to completion of a processing operation.

127. A data structure as in claim 126, wherein said processing operation includes a file system operation.

128. A data structure as in claim 126, wherein said processing operation includes reuse of said selected member storage blocks by said file system.

129. A data structure as in claim 121, wherein said shadow snapshot is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

130. A data structure as in claim 129, wherein said set management operation is a logical sum or difference.

131. In a file system having a plurality of storage blocks, a data structure including a mark-on-allocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been added to a snapshot that originally formed a consistent file system.

132. A data structure as in claim 131, wherein said mark-on-allocate storage image is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

133. A data structure as in claim 131, wherein said mark-on-allocate image is disposed in a format allowing for a set management operation to be performed efficiently.

134. A data structure as in claim 131, wherein said mark-on-allocate storage image uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

135. A data structure as in claim 131, wherein said mark-on-allocate image uses no more than four bytes per storage block in said plurality.

136. A data structure as in claim 131, said member storage blocks having been selected responsive to completion of a processing operation.

137. A data structure as in claim 136, wherein said processing operation includes a file system operation.

138. A data structure as in claim 136, wherein said processing operation includes reuse of said selected member storage blocks by said file system.

A1 Cm.T

139. A data structure as in claim 131, wherein said mark-on-allocate image is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

140. A data structure as in claim 139, wherein said set management operation is a logical sum or difference.

141. In a file system having a plurality of storage blocks, a data structure stored in said file system, including a mark-on-deallocate image of a set of member storage blocks selected from said plurality, said member storage blocks having been removed from a snapshot that originally formed a consistent file system.

142. A data structure as in claim 141, wherein said mark-on-deallocate storage image is disposed as a single object in said file system, whereby said file system can manipulate said snapshot without having to traverse a hierarchy of file system objects within said snapshot.

143. A data structure as in claim 141, wherein said mark-on-deallocate image is disposed in a format allowing for a set management operation to be performed efficiently.

144. A data structure as in claim 141, wherein said mark-on-deallocate storage image uses no more than 1/100th of an amount of storage required by said storage blocks in said plurality.

145. A data structure as in claim 141, wherein said mark-on-deallocate image uses no more than four bytes per storage block in said plurality.

146. A data structure as in claim 141, wherein said mark-on-deallocate image is disposed in a format allowing for a set management operation to be performed in O(n) time or less, where n is a number of storage blocks in said plurality, without reading any contents of said storage blocks in said plurality.

147. A data structure as in claim 146, wherein said set management operation is a logical sum or difference.